The Design Process of an Evolutionary Oriented Reconfigurable Architecture

Ricardo S. Zebulum
Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91109, USA
ricardo@brain.jpl.nasa.gov

Adrian Stoica
Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91109, USA
astoica@brain.jpl.nasa.gov

Didier Keymeulen
Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91109, USA
didier@brain.jpl.nasa.gov

Abstract - This paper describes the design of a reconfigurable chip programmable at the transistor level and oriented to the implementation of Evolvable Hardware (EHW) experiments. We tackle the main issues referring to the conception of an Evolutionary Oriented Reconfigurable Architecture (EORA): the cell topology; interconnection between cells; transistor sizing; resistor and capacitor implementation in silicon; selection of inputs and outputs points; and re-configuration aspects. A set of evolutionary experiments is described, serving as support for the design choices. Additionally, we propose novel approaches to overcome area requirements for the VLSI design, such as the use of differentiated configurable blocks and a variable interconnection density throughout the chip.

1 Introduction

A reconfigurable chip called Field Programmable Transistor Array (FPTA) is currently being designed at the Jet Propulsion Laboratory (JPL). We target the design of a programmable chip that is suitable as a platform for EHW experiments [7]. This kind of project considers many design aspects, such as the cell architecture, interconnection among cells and transistor sizing. We approach this task through the execution of a number of evolutionary experiments whose objective is to find out the best design options for the reconfigurable chip.

The design of Evolutionary Oriented Reconfigurable Architectures (EORA) is distinct from the design of conventional architectures. In addition to contemplating basic VLSI aspects, such as area, layout and power consumption, the design of EORA encompasses the selection of a set of important features for EHW experiments. These features range from the reconfiguration level to the cells interconnections. This paper is devoted to showing how to handle these aspects in the design of a reconfigurable chip.

This work is organized as follows: section 2 overviews existing technology of reconfigurable chips, emphasizing Field Programmable Analog Arrays (FPAAs). Section 3 describes several issues related to the design of the FPTA, including the cell architecture, interconnections, transistor sizing, resistors and capacitors implementation, selection of input and output points, and the chip programming interface. Section 4 presents the new chip architecture and section 5 concludes the work.

2 Overview of Existing Technology

This section reviews current technology on reconfigurable devices, focusing on FPAAs. *Field Programmable Analog Arrays* and *Field Programmable Gate Arrays (FPGAs)* promise to establish a new trend in electronic design, where a single device has now the flexibility to implement a wide variety of electronic circuits. While FPGAs have been developed for applications in the domain of digital signal processing and re-configurable computing, most FPAA models are being developed for applications in programmable mixed-signal circuits and filtering. In addition to the intrinsic flexibility of these devices, which confers advantageous features to standard electronic design, FPGAs and FPAAs are also the focus of research in the area of *self programmable* systems. In EHW, Genetic Algorithms (GAs) are employed to promote the automatic synthesis of electronic circuits over programmable chips.

Most FPGA models consist of an arrangement of cells that perform digital logic, such as basic gates, multiplexers and flip-flops. Many surveys of FPGA models can be found in the literature [1]; therefore, this section focuses on the description of FPAA technologies.

FPAAs are usually regular architectures consisting of a matrix of Configurable Analog Blocks (CABs), also called cells. They are often identical being constituted by one Operational Amplifier (OpAmp) with programmable interconnections. In the present moment, reconfigurable devices have up to 20 cells integrated on the chip.

The design of conventional FPAAs, i.e., those not particularly intended for evolutionary experiments, faces many challenges, the most important ones being the bandwidth, switch resistances, accuracy, noise and area. We now describe current technologies being used to address these problems:

?? **Switched Capacitor (SC) Technology** – This technology is used in the Motorola MPAA0020 FPAA. It confers advantages in terms of accuracy in the frequency response [2]. However, the bandwidth of the circuit is limited by the clock frequency. Another

problem with SC is the lack of simulation accuracy when using standard simulators.

?? **Current Conveyors** – The use of a current conveyors as FPAA cells, instead of OpAmps, allows the bandwidth to increase until about 10 MHz, comparing to 1MHz observed in other models. Current conveyors are very similar to OpAmps and can also implement many different analog sub-systems. The reader can refer to [3] for more details.

?? **BiCMOS Technology** – The use of BiCMOS technology, as proposed in [4], is another way to increase speed and bandwidth of the reconfigurable device.

?? **PulseWidth Technologies** – Proposed in [4], instead of representing signals as voltages or currents, digital pulses are used to represent discrete analog signals. This facilitates the interface with digital systems.

?? **External Components** – In order to minimize the chip area, external capacitors and resistances can be used, instead of integrating them in silicon, as in the Zetex chip [5]. However, this approach reduces the versatility of the reconfigurable device.

?? **Differential Architecture** – This strategy is followed by the Lattice FPAA design [6], and it is effective to improve the performance in terms of input common mode rejection and dynamic range.

?? **Antifuse Switches** – The advantage of using Antifuse technology as switches is the fact that it can provide a lower resistance (around 20 ?) compared to MOS based switches (around 1K), thereby increasing the device bandwidth. The drawback is the fact that antifuses are only one time programmable.

In addition to the above issues, the design of EORA must take into account other factors. Perhaps, the most important of these issues is the one of the granularity of the programmable chip. According to this criteria, programmable devices can be divided into two classes, coarse grained and fine grain devices. While the former uses more complex circuits, such as operational amplifiers, as the configurable blocks, the latter is configurable at a lower level, usually the one of transistors. The technologies presented so far are employed in coarse grained devices, which, as it will be discussed later, is a limiting factor for their application in evolutionary experiments.

3 FPTA Design Process

This session addresses the main issues associated to the design process of the FPTA chip. We describe the cell architecture, input and output selection, cell interconnections, capacitor implementation, resistor implementation, transistor sizing and programming interface. The number of tests needed to contemplate these issues can be prohibitively large, hence our strategy is to handle separately each of these aspects.

3.1 Cell Architecture

We start by describing the cell topology of a first version of the FPTA. This cell is shown in Figure 1. The cell is an array of transistors interconnected by programmable switches. The status of the switches (ON or OFF) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be represented by a binary sequence, such as "1011…", where by convention one can assign '1' to a switch turned ON and '0' to a switch turned OFF. In this implementation, transistors P1-P4 are PMOS and N5-N8 are NMOS, and the switch based-connections are in sufficient number to allow a majority of meaningful topologies for the given transistors arrangement [7].

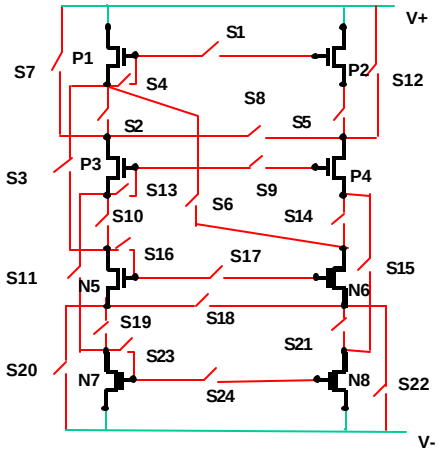

Figure 1 - Schematic of the first version of the FPTA cell consisting of 8 transistors and 24 switches.

The most advantageous feature of this cell in comparison to other programmable devices is the reconfiguration at transistor level. This feature allows definition of building blocks or subcircuits at a variety of levels of granularity. At the lowest level one can configure subcircuits such as current mirrors and differential pairs using only one cell, while more complex blocks, such as logical gates and OpAmps, can also be easily configured with one or two cells. The level of granularity can be set by the designer, who can either let the evolutionary process manipulate the cell at the transistor level, or freeze the cells architecture to well known high level analog or digital building blocks, and let evolution manipulate them. In the former approach, one can expect evolution to come with the building blocks that are the most suitable for the particular application.

The FPTA cell shown in Figure 1 was manufactured in 0.5 micron CMOS technology. The chip allowed us to take circuits obtained through evolution in simulations and validate them by downloading and evaluating their performance in hardware. These tests have enabled us to propose an improved architecture for the next chip design.

3.2 Input and Output Collecting points

A relevant question to be addressed in the design of the FPTA is the selection of input and output application points. In the case of input points, we must investigate how many and which cell points should be accessible for input signals [8]. Two case studies address these two issues.

3.2.1 Case Study I: Number of Input points

In the case of conventional programmable chips, two input points are usually accessible per cell, in order to provide differential operation [6]. In the case of EORA, it has also been verified that evolution needs more than one application point to deliver good results. This concept can be exemplified in one case study where we accomplish the evolution of a computational circuit with a Gaussian output [8]. Two sets of experiments are performed: the first one using four input points (3D, 3G, 5G and 5S)[1] and the second one applying the input to a single point in the cell, the gate of transistor 5 (as in a human designed circuit [8]). The graph of Figure 2 compares the performance of the two experiments, plotting the average fitness of the best individual over four GA executions. Each GA execution sampled 128 individuals along 200 generations. From the comparison made in Figure 2, it can be clearly seen that the GA performance is greatly improved if we allow more than one input point.

The graph of Figure 3 displays the response of the best individual achieved when using multiple inputs.

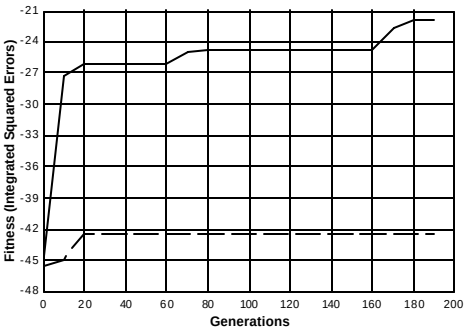

Figure 2 – Comparison between one-input (traces) and multiple inputs (full line) experiment for a Gaussian circuit.

3.2.2 Case Study II: Selection of Input Points

We describe now two sets of experiments whose objective is to evaluate if there are "preferable" input points used by evolution. The experiments refer to the evolution of two different circuits, a computational circuit and a band-pass filter. The computational circuit is more complex than the single Gaussian circuit introduced previously, producing a "double Gaussian" shaped output. The second circuit is a wide band pass filter with passing band between 100kHz and 1MHz. The fitness used to evaluate the computational circuit is given by the MSE to the target specification. In the case of the filter, according to the particular frequency band, the circuit fitness is increased by a specific constant depending on the distance to the target.

We ran four GA executions, processing 200 individuals along 200 generations. We tested eight programmable switches interconnecting the input to one of the FPTA cells, as depicted in Figure 4. The output is collected from cell 2.

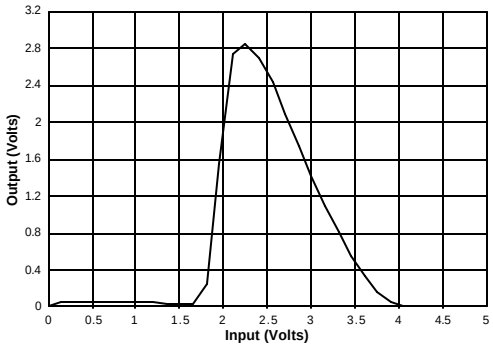

Figure 3 – DC transfer of the best circuit achieved in the multiple inputs experiment.

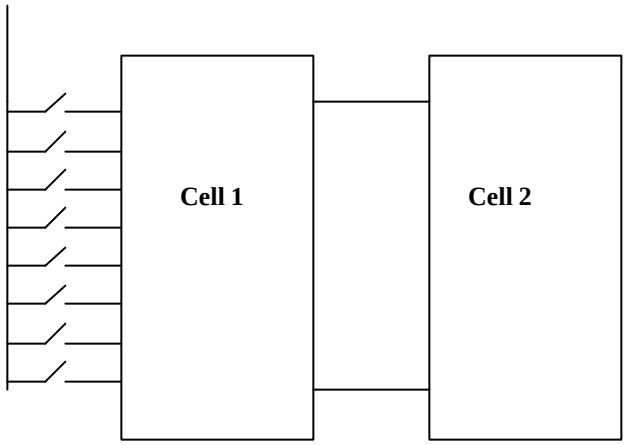

Figure 4 – Topology for the second set of experiments: selection of input points.

The input application points were the following ones: 1D, 3D, 3G, 3S, 5D, 5G, 5S and 7G. Table 1 shows how many times, out of four GA executions, each input switch was turned on for the best individuals achieved in the computational circuit and band pass filter experiments. From this table, one can see that the terminals 3D, 3G, 5G and 5S are more frequently used by the GA. The best individuals have an average of 4 closed switches, for both the computational circuit and the filter. This fact supports the idea of multiple application points for the inputs. The best response achieved for the computational circuit experiment is shown in Figure 5, and for the band pass filter is shown in Figure 6.

[1] Notation gives transistor number and terminal, i.e., 3G, gate of transistor 3 (Figure 1)

Referring to the output probing points, we can select high impedance points (transistor drain or gate) if we desire a circuit operating in the current mode, or a low impedance point (transistor source) if we need a buffered output. As we will show in section 4, we will let four different points available for the cell output.

Table 1 – Frequency at which the input programmable switches were turned on in case study II (3.2.2).

Input Points	Computational Circuit	Band-Pass Filter
1D	1/4	2/4
3D	3/4	4/4
3G	2/4	3/4
3S	3/4	0/4
5D	0/4	1/4
5G	3/4	3/4
5S	3/4	3/4
7G	3/4	2/4

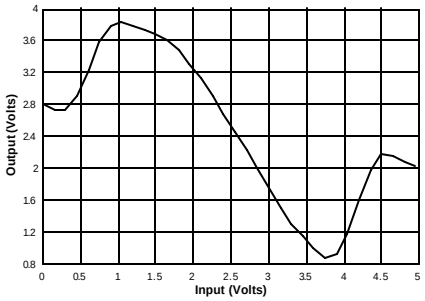

Figure 5 – Response of the best computational circuit (Selection of input points).

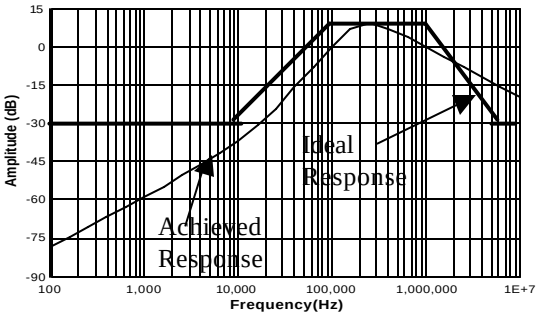

Figure 6 – Response of the best bandpass filter. (Selection of Input points). Axis X gives frequency in Hz.

3.3 – Cell Interconnections

The number and the kind of connections between cells is another important design decision to be made. Naturally, this parameter depends on the number of inputs/outputs available in the cells. We present two case studies in this section: the first one is related to the number of interconnections, and the second one is related to the kind of interconnections.

3.3.1 – Case Study I: Number of Interconnections

We compare the performance of the evolutionary algorithm when two and four fixed connections are applied between two cells. The connections are labeled as I1 ($4D_a$-$5G_b$), I2 ($6S_a$-$6G_b$), I3 ($8D_a$-$5S_b$) and I4 ($1D_a$-$2D_b$) in Figure 7, where *a* is the cell in the left and *b* is the cell in the right. We then compare two cases: when I1, I2, I3 and I4 are used; and when only I1 and I2 are used.

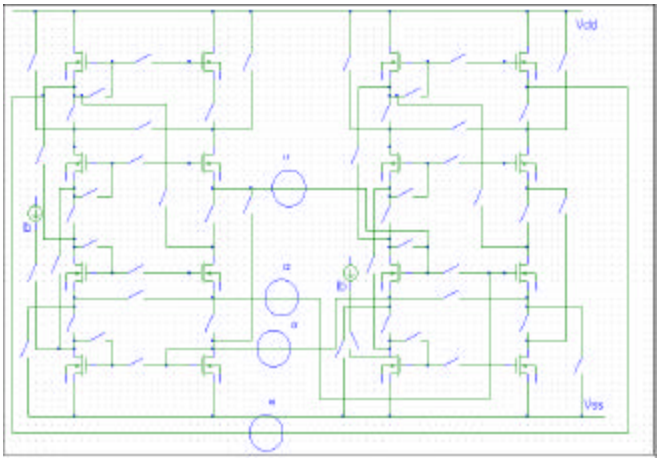

Figure 7 – Circuit topology utilized in the first case study (Number of Interconnections).

We performed experiments with a computational circuit (simple gaussian) and with the previously defined wide band pass filter. Figure 8 compares the performance of the Gaussian circuit evolution using two and four connections. We performed three GA executions sampling 200 individuals along 250 generations. From the graph, it can be verified that using four connections slightly improved the performance.

It is interesting to note, however, that the situation is the opposite for the bandpass filter, i.e., the average performance using two connections is slightly superior than when using four connections. This is shown in the graph of Figure 9.

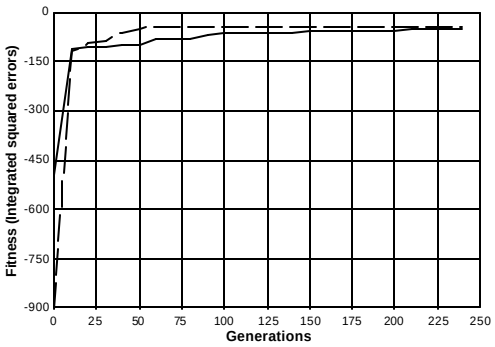

Figure 8 – Comparison of the average fitness achieved in the Gaussian experiment: two connections (full line) and four connections (traces).

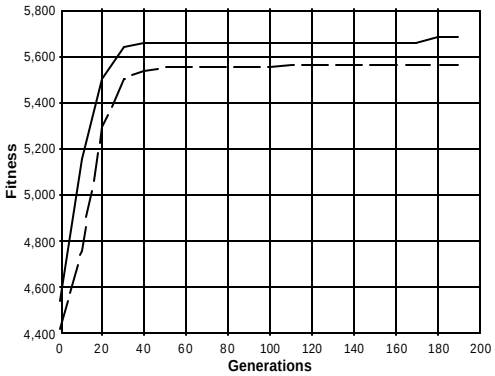

Figure 9 - Comparison of the average fitness achieved in the Bandpass filter experiment: two connections (full line) and four connections (traces).

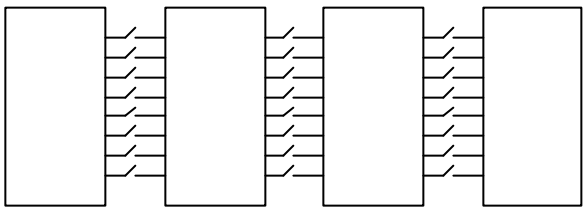

Figure 10 – Circuit Topology for the case study II (Type of Interconnections).

3.3.2 Case Study II: Type of Interconnections

Similar to the case of selection of input points, it is also interesting to investigate if "preferable" interconnections between cells exist. We used as target circuit, in this case study, the evolution of the two-Gaussian computational function. The circuit topology consisted of four cells connected through eight programmable interconnections. The objective is to make a statistical analysis of which connections might be more relevant to the synthesis of more complex computational functions. Figure 10 depicts the circuit topology used in the experiments.

The following interconnections have been used: 4D-5G; 6S-6G; 2D-1D; 8G-5S; 2G-2G; 4S-4S; 5D–6S; and 1G-3D. Note that, in this case, the two connecting terminals belong to adjacent cells. Different connection types were tested, i.e., D-D, D-G, D-S, G-G, G-S, and S-S. We ran four executions, each one sampling 200 individuals along 200 generations. Table 2 shows the outcome of the experiment. This table indicates how many times each switch was turned on over a total of 12 cases (each switch appears three times in the topology, being multiplied by a total of 4 experiments). From this table it can be verified that the best individual used equally the different types of connections.

Table 2 – Rate of switches turned on in the second case study (Type of Interconnections).

Connection	Switch ON
4D-5G	6/12
6S-6G	4/12
2D-1D	4/12
8G-5S	6/12
2G-2G	4/12
4S-4S	7/12
5D-6S	4/12
1G-3D	5/12

3.4 Capacitor Implementation

One limitation of the FPTA topology presented in Figure 1 is the absence of capacitors. Although that model is suitable for the evolution of computational circuits, as shown in the previous experiments, capacitors are required for the synthesis of other kind of analog circuits, mainly filters.

The main drawback of integrating capacitors on chip is the increase in the amount of chip area, this being the reason why some FPAAs use external capacitors. As it will be described in section 4, we propose a compromise, by restricting the availability of capacitance resources to a few cells of the chip. In our previous experiments on the wide band filter evolution, we included programmable capacitors in the simulated topology, as depicted in Figure 11. In this figure, the programmable capacitors are implemented by a parallel arrangement of three capacitors, each one in series with a switch.

There are two design decisions to be made for on-chip capacitor implementation: the position of the capacitors and the possible values they can assume. Referring to the topology presented in Figure 11, two programmable capacitors were used, the first one in the input of the first cell and the second one in the output of the second cell. They respectively create the zero and the pole for the band-pass filter. It can also be observed from the figure that the capacitor array consists of three parallel components,

measuring 0.1n, 1n and 10n respectively. Each programmable capacitor can then assume 2^3 different values, according to the switches configuration, and ranging from 0.1 to 11n in this case. Note that the output capacitor is in parallel with the circuit load, which is approximately 0.1n.

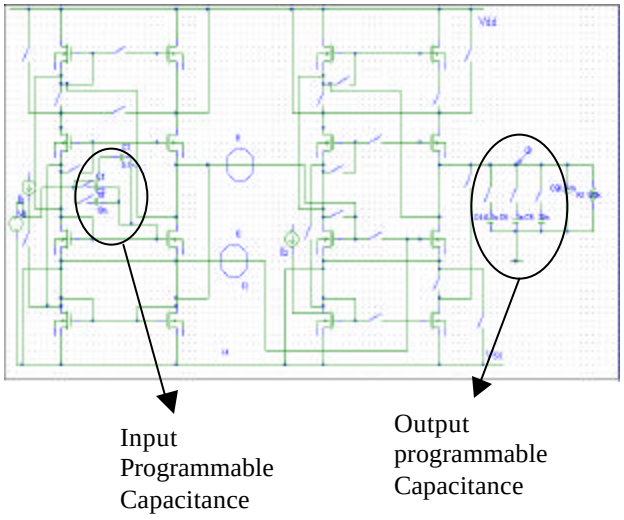

Figure 11 – Circuit topology using programmable capacitors.

3.5 Resistor Implementation

The on-chip implementation of resistors will endow the configurable cell with more versatility for the implementation of analog circuits. Currently, there are two approaches for resistor implementation: using a programmable resistor array or using switched capacitors. The former is very similar to the one described in the previous section for programmable capacitors, exchanging the capacitors for resistors. This approach is described in [9], where six parallel resistors are used, providing minimum and maximum values of 10k and 320k respectively. The second approach, switched capacitors, increases the accuracy of analog filter design. However, both of these approaches have the drawback of increasing the chip area.

The approach proposed in this paper is to explore the resistive properties of the programmable switches. These switches are implemented through transmission gates (t-gates) and we can control their states to achieve partly opened / partly closed configuration, where they present intermediate resistor values. Figure 12 presents the schematic of the transmission gate and a graph showing the switch resistance as a function of the gate control voltage (ignoring effects of parasitic capacitances). It can be seen from this graph that the opened switch corresponds to a value of 10^{12} ? , while the closed switch corresponds to a value of 1000? . If we use intermediate control voltages values (between 0V and 5V), then we can achieve intermediate resistance values, as shown in the graph.

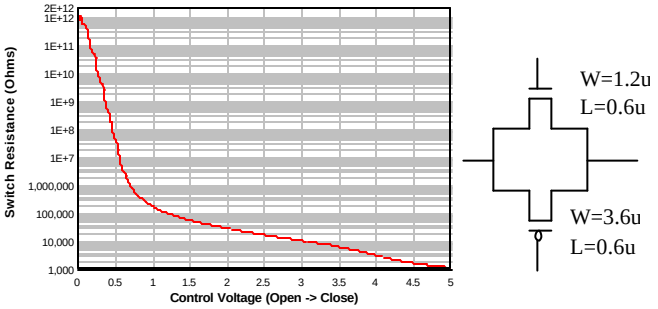

Figure 12 - Schematic of the transmission gate and resistance values as a function of the control voltage (Axis Y expressed in Ohms and axis X in Volts).

The following comparative experiment was implemented to test three different approaches for resistance implementation: using resistors with values and connecting points programmed by the GA; using no resistors and t-gate switches completely opened or closed; and using no resistors with partly opened/closed t-gate switches. In the last test we used control voltage values of 1.5 and 3.5 Volts to attain intermediate resistance values. The graph of Figure 13 shows the results for the two-gaussian computational function. It can be verified that the use of partially opened/closed switches greatly improves the GA performance for this particular problem. The graph shows the average population fitness over 4 GA executions that sampled 40 individuals along 100 generations.

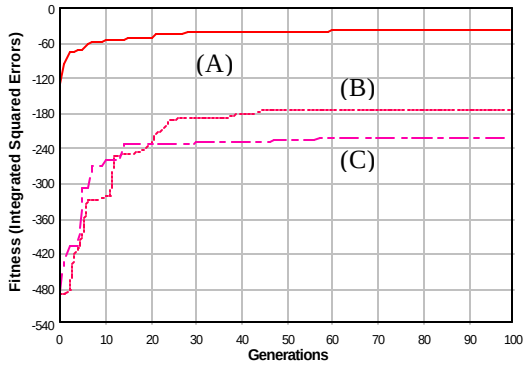

Figure 13 – Comparative experiment with programmable resistance: (A) – partly opened and closed switches; (B) – completely opened and closed switches; (C) – using resistors.

3.6 Transistor Sizing

The behavior of CMOS circuits depends not only upon the topology, but also on the transistors sizes, width (W) and length (L). As it will be shown in section 4, the chip will be organized in cells clusters with different values of W/L. This will allow an additional degree for performance optimization.

3.7 Programming Interface

The programming interface consists of the digital circuitry that receives the chromosome bitstring and configure the switches. It is basically constituted of a buffered shift register of size *n*, where *n* is the number of bits of the chromosome. One important feature to increase the efficiency of the process is the capacity of partial reconfiguration, which allow the user to reconfigure only some FPTA cells, whilst keeping the state of the other ones. Additional control logic should then be provided to include this feature.

4 Chip Architecture

Let us describe now the chip architecture. We start presenting the architecture at the cell level. Following we present the overall chip architecture.

The new cell architecture is presented in Figure 14. This topology keeps the 24 internal switches, adding four additional switches, S25, S26, S27 and S28. These switches offer different application points for the input signal, going into the terminals 3D, 3G, 5G, 5S. The outputs may be collected from four different points, 2D, 4D, 6S and 8G. This new topology results from the experimental evidence of the advantages of multiple input/output points.

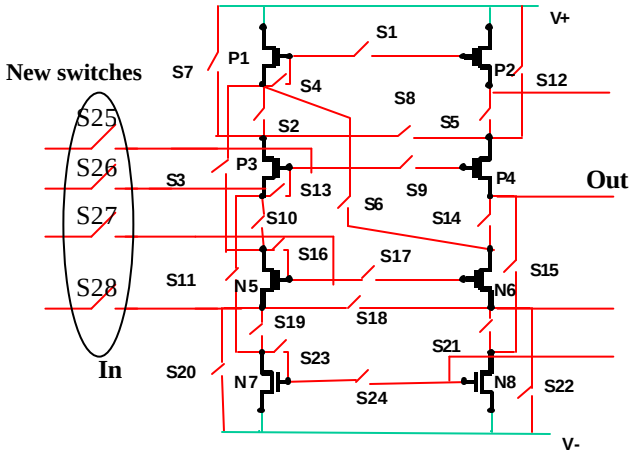

Figure 14– New FPTA Cell. (Inputs in the left, outputs in the right).

One of the advantages of this architecture is the fact that there is no illegal bitstring, i.e., one that would damage the cell by short circuiting power to ground for instance.

The problem of having cells with many inputs/outputs is the increase in the routing complexity, particularly if we need to achieve a high density of interconnections among cells. In order to overcome this problem, we connect the four input points two by two, reducing the number of I/O points from 4 to 2. This is shown in Figure 15. According to the schematic shown in this figure, we connect two external inputs, *In1* and *In2*, to the four input points. Referring to the outputs, two 2-inputs analog multiplexers map the four output points to two external outputs *Out1* and *Out2*. The bits represented by S29 and S30 control the two multiplexers. Using this architecture, each cell will be configured by 30 bits (topology + interconnections).

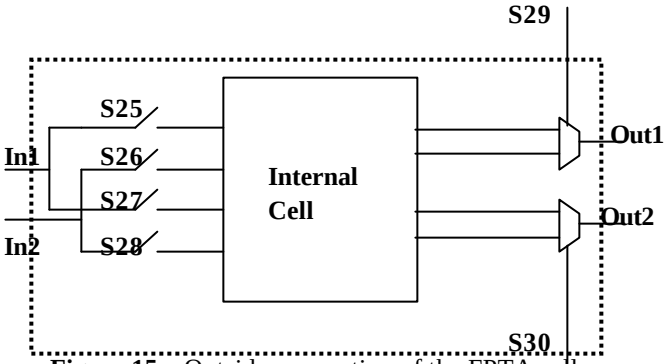

Figure 15 – Outside connection of the FPTA cells.

Let us now present the architecture at the chip level. The chip will consist of a 6x6 matrix of cells. This is shown in Figure 16.

We can divide the cell matrix into boundary cells and central cells. The boundary cells are in a number of 20. The boundary cells receive the external input signals *In1* to *In20*. These cells can be optionally used just to pass the input signals to the central cells or to route the output signals to the output buffers (a total of 10 output buffers). All the boundary cells have their outputs going to the central cells, except for the vertex cells CB1, CB2, CB3 and CB4, whose output go to their neighbors, (CB5 to CB12) as shown in the figure.

Figure 17 depicts the interconnectivity of the central cells, expanding the dashed rectangle shown in Figure 16. Each cell can receive, as inputs, the outputs of their neighbor cell to the north, south, east and west, according to the selection bits of a 4-1 analog multiplexer. As there are two inputs and two outputs per cell, two analog multiplexers are used. Therefore, the density of interconnections is higher for the central cells comparing to the boundary cells.

The eight inner cells are the only ones that present programmable capacitors, implemented in the way described in section 3.4. There will be two programmable capacitors per cell, connected to its input and output respectively.

As previously mentioned, the use of different W/L ratios for the MOS transistors confers to the GA an additional dimension for optimization. We then divide the chip into 9 clusters of 4 cells, presenting the following W/L ratios in ?: 1.2/1.2; 2.4/1.2; 4.8/1.2; 2.4/2.4; 4.8/2.4; 9.6/2.4; 4.8/4.8; 9.6/4.8; 19.2/4.8.

Finally, Table 3 overviews some important statistics of the chip.

5 Conclusions

This work presented the design process of an Evolutionary Oriented Reconfigurable Architecture, the FPTA. We discussed the main features of currently available reconfigurable chips, emphasizing FPAAs. A number of evolutionary experiments to support some design decisions for the FPTA was described. Based on these experiments, the chip architecture was then presented. This chip is going to be submitted by the time of publication of this paper.

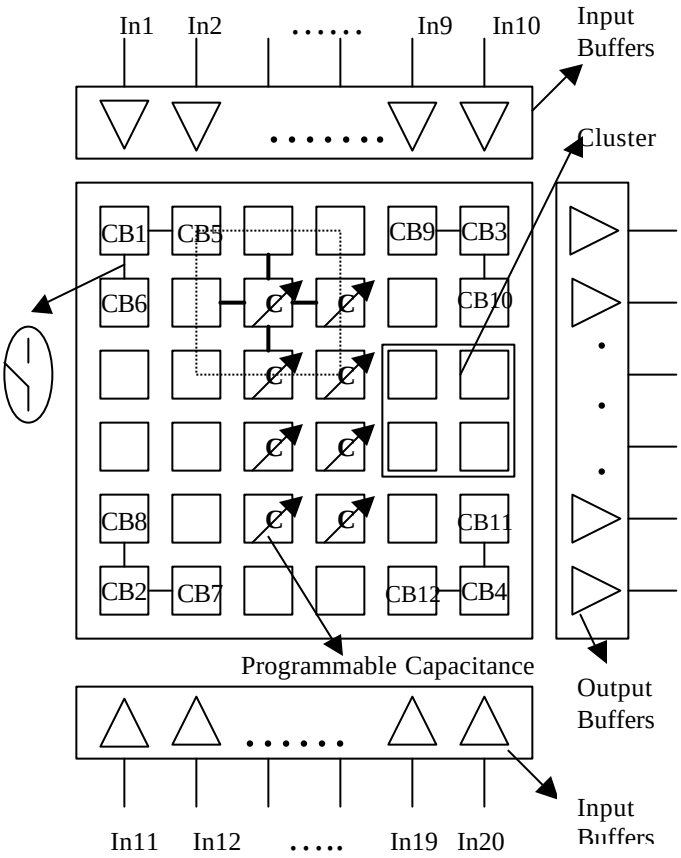

Figure 16 – Global view of the reconfigurable chip.

Table 3 – Overall chip statistics.

Configuring bitstring	Around 1200 bits
Number of Inputs	20
Number of Outputs	10
Number of pins (Estimated)	80

Acknowledgements
This research was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Inst. of Technology and was sponsored by the Defense Advanced Research Projects Agency (DARPA) under the Adaptive Computing Systems Program.

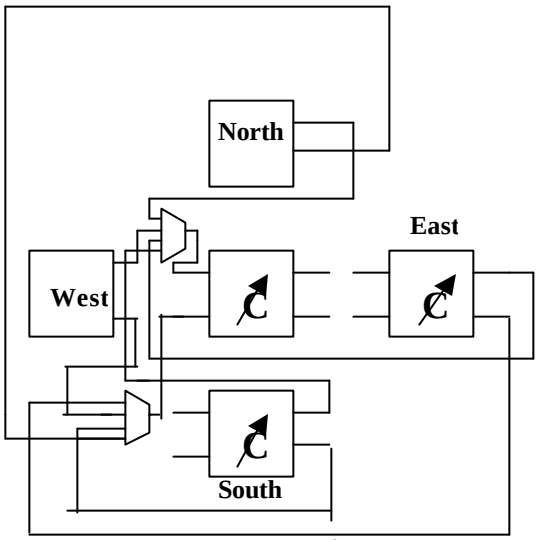

Figure 17 – Interconnections for the central cells. (cell inputs in the left, outputs in the right).

References:

[1] - Sanchez, E. (1996). "Filed Programmable Gate Array (FPGA) Circuits", in *Towards Evolvable Hardware*, pp. 1-18, E. Sanchez and M. Tomassini (editors), Springer-Verlag LNCS 1062.

[2] - Johns, D. A and Martin, K. (1997). "Analog Integrated Circuit Design", John Wiley and Sons.

[3] - Gaudet, V. C., Gulak, P. G. (1997). "Implementation of a Current Conveyor-Based Field Programmable Analog Array"., 31st ASILOMAR Conference on Signals, Systems, and Computers, Pacific Grove, CA, November.

[4] - Hamilton, A., Papathanasiou, K., Tamplin, M. R., Brandtner, T. (1998). "PALMO: Field Programmable Analog and Mixed-Signal VLSI for Evolvable Hardware", in Proc. of the Sec. Int. Conf. on Evolvable Systems. M.Sipper, D.Mange and A. Pérez-Uribe (editors), vol. 1478, pp. 335-344, LNCS, Springer-Verlag.

[5] - Flockton and Sheehan (1998). " Intrinsic Circuit Evolution Using Programmable Analog Arrays", in Proc. of the Sec. Int. Conf. on Evolvable Systems, M. Sipper, D.Mange and A. Pérez-Uribe (editors), vol. 1478, pp. 144-155, LNCS, Springer-Verlag.

[6] - Lattice Semiconductor Corporation, (1999). "ispPAC Handbook, Programmable Analog Circuits", Sep., 1999.

[7] - Stoica, A., Zebulum, R. S., Keymeulen, D., Tawel, R., Taher, D., Thakoor, A. (2000). "Reconfigurable VLSI Architectures for Evolvable Hardware:from Experimental Field Programmable Transistor Arrays to Evolution-Oriented Chips", in review by the IEEE Trans. on VLSI.

[8] - Stoica, A., Keymeulen, D., Tawel, R., Salazar-Lazaro, C., Li, W. (1999). "Evolutionary Experiments with a fine-grained reconfigurable architecture for analog and digital CMOS circuits", Proc. of the First NASA DoD Workshop on Evolvable Hardware, pp.76-84, IEEE Computer press., July.

[9] - Edwards, R. T., Strohbehn, K., Jaskulek, S. E. (2000). "A Field-Programmable Mixed-Signal Array Device", John Hopkins University Applied Physics Laboratory, lecture notes, Feb. JPL.